

What is claimed is:

1. A method of transferring data, comprising:

5 completing a first data transfer;

determining that no data transfers are being processed;

and

releasing a second data transfer for processing.

10 2. The method of claim 1 wherein the second data transfer includes two or more combined transfer requests.

3. The method of claim 2, further comprising:

determining there are pending transfer requests to

15 coalesce before releasing the combined data transfer.

4. The method of claim 3, wherein releasing comprises determining that the combined data transfer is an optimum-size that corresponds to an address boundary of an input/output 20 device.

5. The method of claim 4, wherein the address boundary corresponds to a strip boundary corresponding to a redundant array of inexpensive disks (RAID) process.

6. The method of claim 3, further comprising:
storing pending transfer requests; and
determining that the number of pending transfer requests
5 stored is not greater than a queue depth variable before
storing a new transfer request.

7. The method of claim 5, further comprising:
gathering statistics corresponding to the performance of
10 the method of claim 3; and
modifying the queue depth variable based on the gathered
statistics.

8. The method of claim 3, wherein a second transfer
request is stored on the pending list, and
wherein the second transfer request is not addressed
15 adjacent to the first transfer request address, and
wherein a third transfer request is coalesced with the
first and second transfer requests into the combined data
transfer, the third transfer request being adjacent to the
20 first and second transfer requests.

9. A method of transferring data, comprising:

receiving transfer requests;
releasing data transfers for processing; and
releasing an additional data transfer for processing each
time a first data transfer completes.

5

10. The method of claim 10, further comprising:
determining there are pending transfer requests, wherein
the additional data transfer comprises a combined data
transfer that includes two or more transfer requests.

10

11. The method of claim 10, further comprising:
determining there are pending transfer requests to
coalesce before releasing the combined data transfer.

15

12. The method of claim 4, wherein releasing the
combined data transfer comprises determining that the combined
data transfer is an optimum-size that corresponds to an
address boundary of an input/output device.

20

13. An article comprising a machine-readable medium that
stores machine-executable instructions for transferring data,
the instructions causing a machine to:
release a first data transfer for processing;

complete a first data transfer;
determine that no data transfers are being processed;
release a second data transfer for processing.

5 14. The article of claim 13, further comprising
instructions causing a machine to combine two or more transfer
requests into a combined data transfer, the second data
transfer being the combined data transfer.

10 15. The article of claim 14, wherein instructions
causing a machine to release comprises instructions causing a
machine to determine that the combined data transfer is an
optimum-size that corresponds to an address boundary of an
input/output device.

15 16. The article of claim 15, wherein the address
boundary corresponds to a strip boundary corresponding to a
redundant array of inexpensive disks process.

20 17. The article of claim 15, further comprising
instructions causing a machine to:
store pending transfer requests; and

determine that the number of pending transfer requests stored is not greater than a queue depth variable before storing a new transfer request.

5 18. The article of claim 17, further comprising instructions causing a machine to:

gather statistics corresponding to the performance of the article of claim 13; and

10 10 modify the queue depth variable based on the gathered statistics.

15 15 19. An apparatus for coalescing transfer requests, comprising:

a memory that stores executable instructions; and

15 15 a processor that executes the instructions to:

release a first data transfer for processing;

complete a first data transfer;

determine that no data transfers are being processed;

20 20 release a second data transfer for processing.

20 20 20. The apparatus of claim 21, wherein the processor executes instructions to:

combine two or more transfer requests into a combined data transfer, the second data transfer being the combined data transfer.

5 21. The apparatus of claim 20, wherein the processor executes instructions to:

 determine there are no pending transfer requests to coalesce before releasing the combined data transfer

10 22. The apparatus of claim 21, wherein releasing comprises determining that the combined data transfer is an optimum-size that corresponds to an address boundary of an input/output device.

15 23. The apparatus of claim 22, wherein the address boundary corresponds to a strip boundary corresponding to a redundant array of inexpensive disks process.

20 24. The apparatus of claim 22, wherein the processor executes instructions to:

 store pending transfer requests; and

determine that a number of stored transfer requests is not greater than a queue depth variable before storing a new transfer request on the pending list.

5 25. The apparatus of claim 22, wherein the processor executes instructions to:

gather statistics corresponding to the performance of the method of claim 23; and

10 modify the queue depth variable based on the gathered statistics.

26. The apparatus of claim 25, wherein the processor executes instructions to:

store a second transfer request on the pending list, wherein the second transfer request is not addressed adjacent to the first transfer request address; and

15 coalesce a third transfer request with the first and second transfer requests into the combined data transfer, the third transfer request being adjacent to the first and second transfer requests.